Thirty Years of Formal Methods Research At NASA Langley

Ricky W. Butler

April 2018

http://shemesh.larc.nasa.gov/fm/

Goal of this talk

History of our FM research

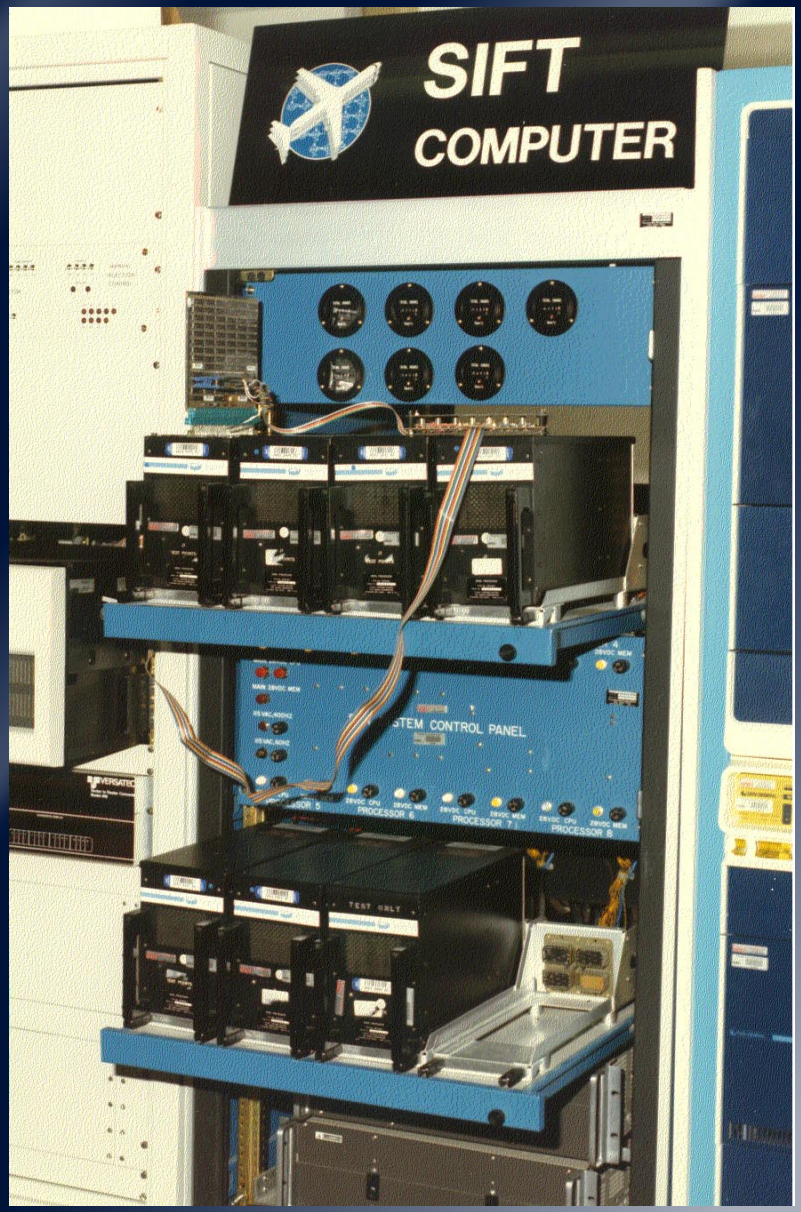
Some lessons learned

NASA Langley Beginnings

- I began work at NASA Langley in the Fall of 1980
- I was the first person hired in my branch who had majored in computer science
- There was a contract in place with SRI International and Bendix to build a fault-tolerant computer named SIFT: Software Implemented Fault Tolerance
- SRI International convinced Nick Murray (SIFT COTR) that there should be a separate contract to formally verify the SIFT operating system [The first NASA funded formal methods project].
- Nick was over-worked, so the branch head decided to transfer the formal verification contract to me in its last year.

SIFT Computer

- Reliability goal: 10^{-9}
- 6 processors
- Fully-connected topology
- Fault-tolerant clock synchronization
- Byzantine agreement algorithm
- Delivered to NASA Langley in 1981
- Contributers include: Jack Goldberg, Chuck Weinstock, Karl Levitt, Michael Melliar-Smith, Richard Schwartz, Rob Shostak, Bob Boyer, Jay Moore, John Wensley, Leslie Lamport

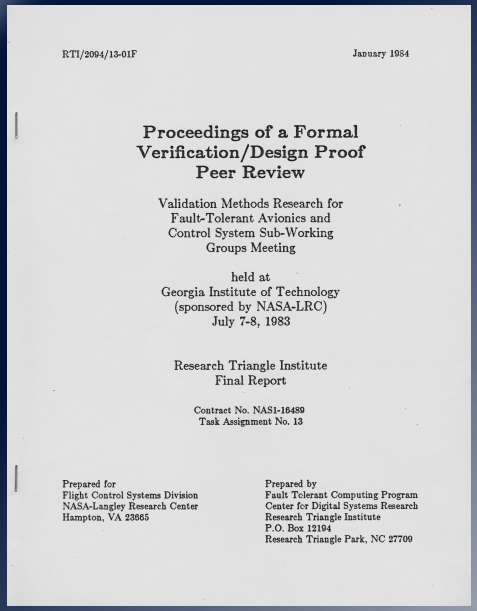


THE SIFT OPERATING SYSTEM VERIFICATION

My branch head (Bill Dove) was very skeptical about formal verification and so he directed me to have a peer review of the project.

He wanted Richard DeMillo (a leading skeptic) to chair the peer review.

The committee gave **a very harsh review** when it learned that the much of the verification was incomplete.



RTI/2094/13-01F January 1984

Proceedings of a Formal Verification/Design Proof Peer Review

Validation Methods Research for Fault-Tolerant Avionics and Control System Sub-Working Groups Meeting

held at
Georgia Institute of Technology
(sponsored by NASA-LRC)
July 7-8, 1983

Research Triangle Institute
Final Report

Contract No. NAS1-15489
Task Assignment No. 13

Prepared for
Flight Control Systems Division
NASA-Langley Research Center
Hampton, VA 23665

Prepared by
Fault Tolerant Computing Program
Center for Digital Systems Research
Research Triangle Institute
P.O. Box 12194
Research Triangle Park, NC 27709

RA DeMillo, RJ Lipton, AJ Perlis, Social Processes and Proofs of Theorems and Programs, 1977 (6th ACM POPL)

Proceedings of a Formal Verification/Design Proof Peer Review (1983)

- "Many publications and conference presentations concerning SIFT appear to have misrepresented the accomplishments of the project"
- "The effort has been contaminated by unfortunate overstatements published in various papers."
- "SRI has not produced a methodology for determining by deductive analysis quantitative measures of fault-tolerance"

The Irony

- Although SRI failed to meet the intent of the contract, i.e. to verify the SIFT operating system
- Some landmark accomplishments had been made:
  - Fault-tolerant clock synchronization
  - Byzantine Agreement
  - An insightful problem decomposition:
    - Prob[enough hardware] via Markov analysis
    - enough hardware ---> good answers
    - Hierarchical decomposition
  - Shostak decision procedures --> Ehdm prover

My Assessment: There was no malice here--SRI was well-intentioned, but the problem was orders of magnitude harder than they had ever imagined

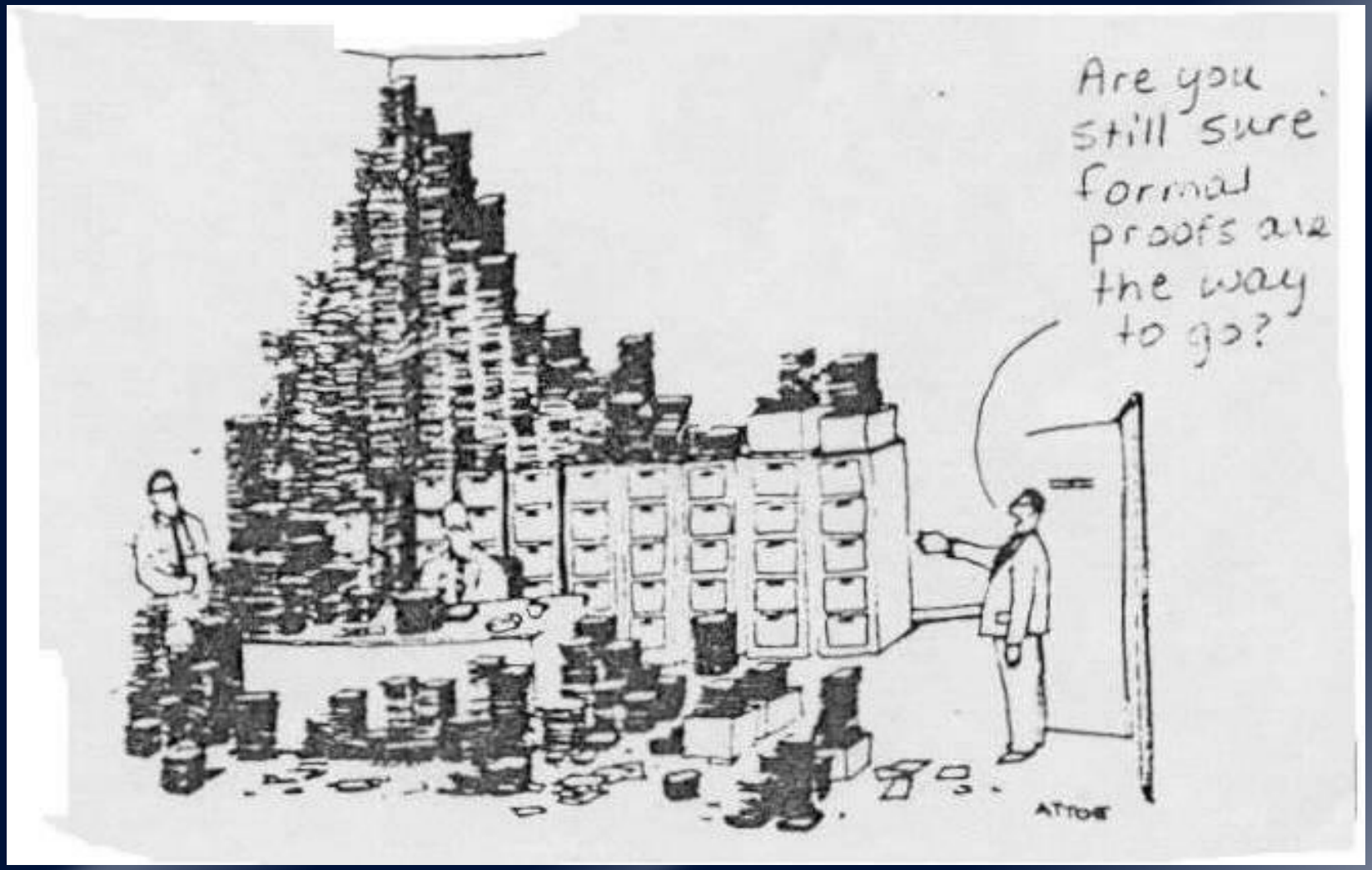
Why Did the Project Fail to meet Its Primary Goal?

- Extremely talented, hard-working team, but they drastically under-estimated the difficulty of the challenge:
  - Fault-tolerance algorithms extremely challenging
  - Goal encompassed verification from requirements to code-level including assembly-level interrupt handler
  - Half-way through project switched tools: Boyer-Moore TP to Shostak Theorem Prover (EHDM)
  - Too many different thrust areas were pursued: tool development, microprocessor modeling, code-level proof, hierarchical specification, design proof, algorithm design proof, etc. etc.
  - The project was run like a university department. It needed a closely cooperating team with a strong leader.

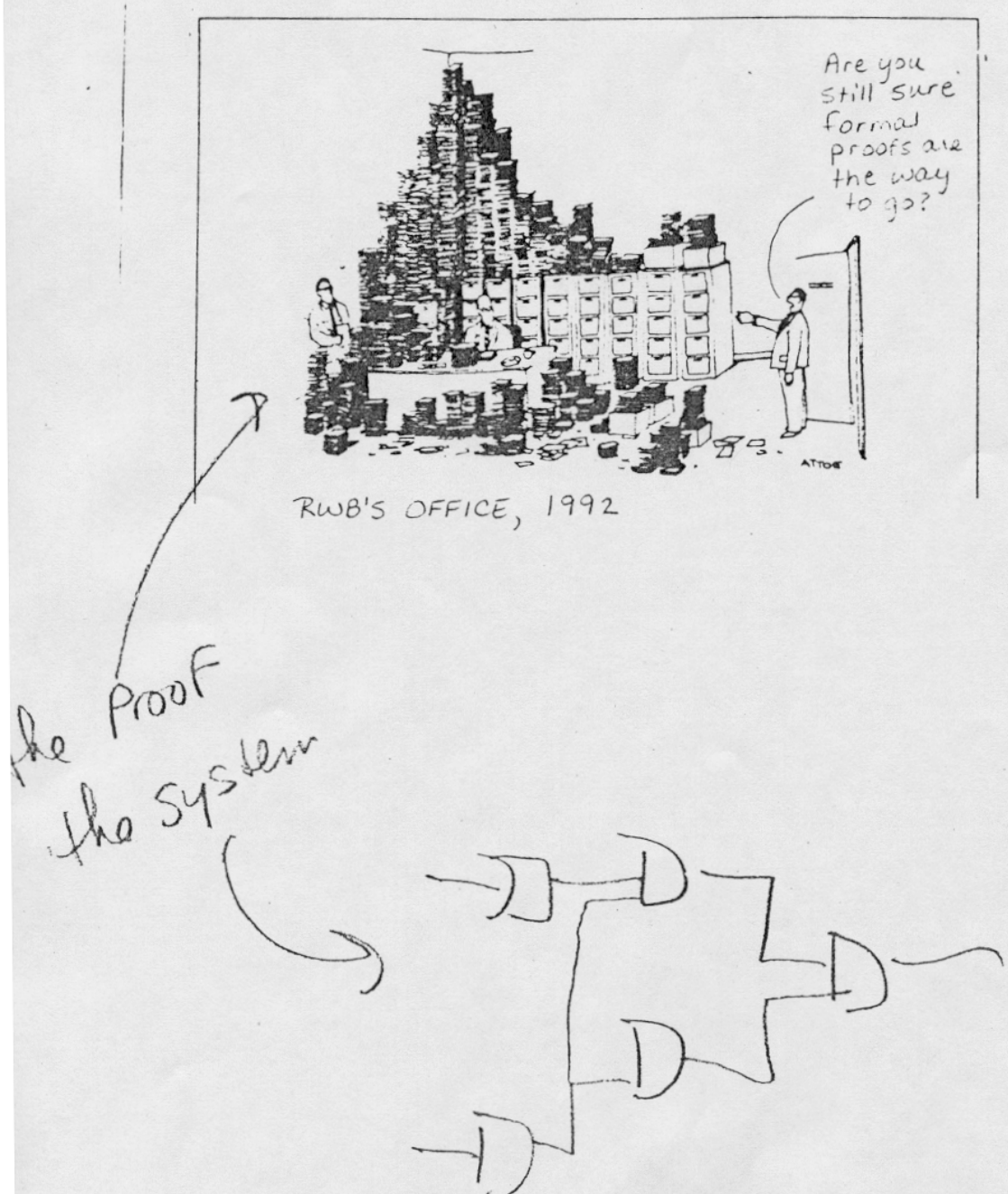
SIFT Operating System Verification

- Final Report: NASA CR 166008, August 1983
  - Ch 2-3: STP (Shostak) theorem prover
  - Ch 7: Design verification of SIFT
  - Ch 9: HDM-Pascal Code Verification System
  - Ch 14: Verification of SIFT Code
  - Ch 17: An initial approach to Verifying a Scheduler--written in Assembly Language
  - Ch 18: Formal Definition of BDX930 Instruction Set
  - Ch 19: Verification of Numerical Algorithms
  - Ch 21: Verification of Hardware Logic
  - Ch 22: Boyer-Moore Theorem Prover

Some Perspective on the Attitude Towards Formal Methods in the Early 1980s



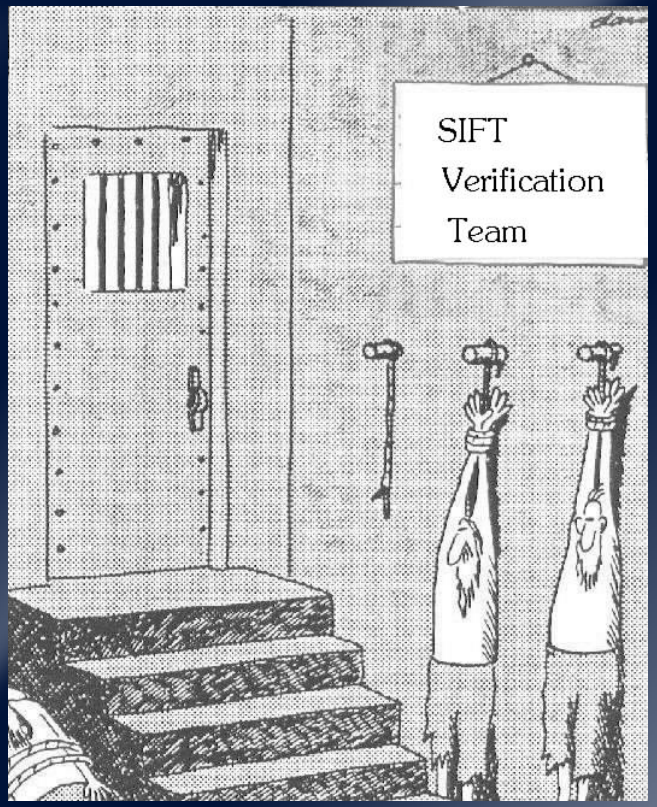
Joke posted in our coffee room



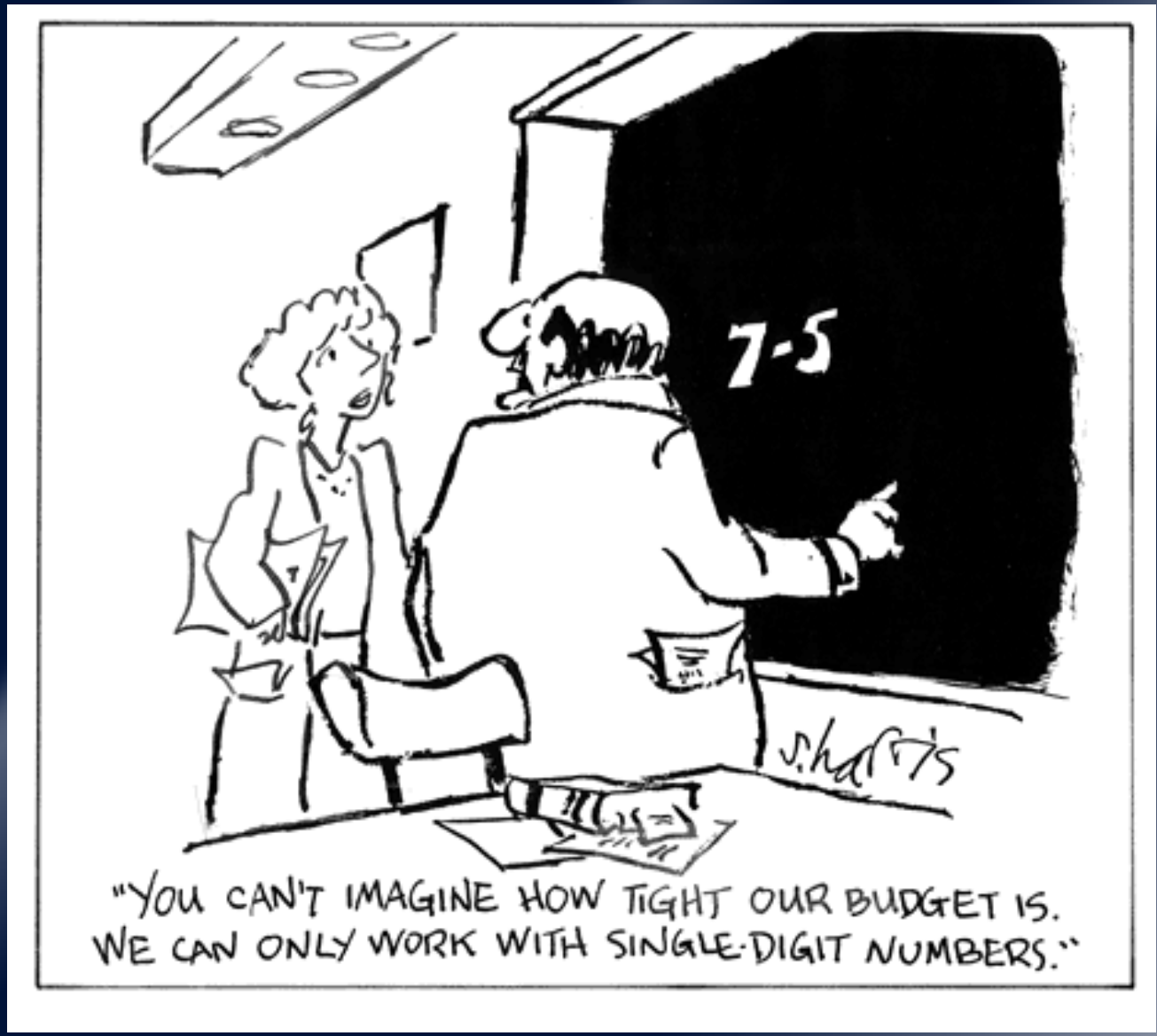
Dan Palumbo's addition

Consequences

- There was some follow-on research dollars to close some of the gaps in the proofs, e.g. reconfiguration, the fault-tolerant clock synchronization algorithm
- But a few years later the funding for formal methods research was 0.



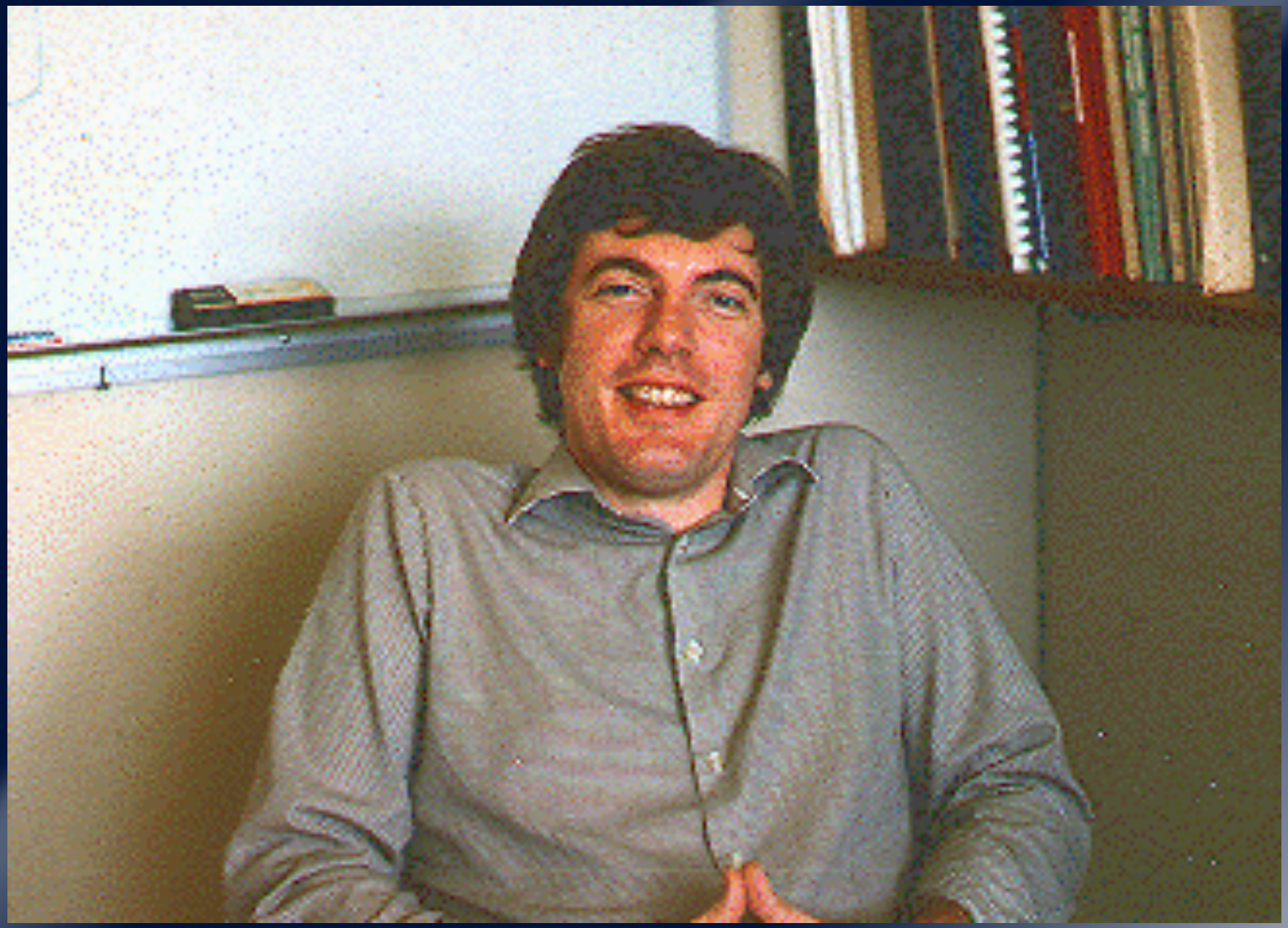
- Rob Shostak and Richard Schwartz left SRI and started a small company named Ansa that created the Paradox database tool. Later Borland acquired Ansa and later sold Paradox® to Corel.


7-5
J. Harris
"YOU CAN'T IMAGINE HOW TIGHT OUR BUDGET IS. WE CAN ONLY WORK WITH SINGLE-DIGIT NUMBERS."

When I first met John Rushby

- In 1985 we funded SRI to formally verify the clock synchronization algorithm of SIFT
- The Principle Investigator had recommended that EHDM be enhanced with decision procedures for an interval temporal logic (that he had invented).
- He had convinced me this was the way to go--this would make the clock sync proof much easier.
- At the end of this task, John Rushby showed up one day in my office.
  - He was now in charge of this work
  - All of the money had been spent!
  - Proofs of clock synchronization algorithm had not been done, interval logic not in Ehdm

This is what John Rushby Looked Like Then



When I first met John Rushby (cont.)

- John Rushby apologized (even though he personally had done nothing wrong)
  - He estimated that the statement of work would take at least 1 million dollars to complete
  - I told him that I didn't care about the interval logic---it was just a means to the end.
  - He volunteered to perform the mechanical proof of the algorithm *free of charge to NASA*
  - *And he succeeded:*

Formal Verification of the Interactive Convergence Clock Synchronization Algorithm. John Rushby and Frieder von Henke, NASA CR-4239, June, 1989.

WHAT WAS EHDM LIKE?

Family: CS_osauria,**Genus:** Stego_prover,**Species:**SRI_auridae

```
stay_correct_repl_proof PROVE stay_correct_repl FROM
   limited_induction
     {p <- (LAMBDA q  rrun(q)(j)(a) = runto(a)(j)(a)),
      m <- dowhen(a),
      m1 <- dowhen(c),
      n <- dowhen(previous(c))},
   r_indstep {m <- i@p1},
   sched_when_lemma {a <- previous(c)},
   when_sched_lemma {m <- pred(dowhen(c))}
```

```
final_proof PROVE the_result FROM
    mod_induction {A <- safe, B <- correct, d2 <- a@p3},
    safe {a <- d4@p1, c <- d3@p1},
    inductive_step {c <- d1@p1}
```

WHAT WAS EHDM LIKE? (cont)

```
Mixed trace for proof the_proof from module sum
   - of which the result was unproved

The conclusion is:
   *q >= 0 AND *q > 0 IMPLIES 2 * sigma(*q) = *q * *q + *q

Premise number 1 is:
   (2 * sigma(1) = 1 * 1 + 1
     AND *m() >= 0 AND *m() > 0
       IMPLIES 2 * sigma(*m()) = *m() * *m() + *m()
         IMPLIES 2 * sigma(*m() + 1) = (*m() + 1) * (*m() + 1) + *m() + 1)
     IMPLIES 2 * sigma(*q) = *q * *q + *q

Premise number 2 is:
   2 * sigma(1) = 1 * 1 + 1

Premise number 3 is:
   2 * sigma(*m()) = *m() * *m() - *m()
     IMPLIES 2 * sigma(*m() + 1) = (*m() + 1) * (*m() + 1) + *m() + 1
```

The Mechanical Proof of a Fault Tolerant Clock Sync Algorithm

This success got us back into the formal verification business!

Because of a clock synchronization failure with fifth backup computer, they had to scrub the first launch attempt on April 10, 1981.

A software patch was installed prior to the next attempt.

Launch of STS-1

Arrival Of John Cullyer in 1987

- But our funding would have remained small except for the arrival of a charismatic researcher from the Royal Signals and Radar Establishment, Malvern England: John Cullyer
- Bill Dove (now the Assistant Division Chief) was extremely impressed by him.
- Cullyer talked about the successful application of the MALPAS toolset to fighter aircraft and
- The VIPER microprocessor that they were developing and formally verifying for military use

---

VIPER= Verifiable Integrated Processor for Enhanced Reliability

RSRE/NASA Langley Collaboration

Dr. C.H. Pygott, Dr. W.J. Cullyer, Dr. J. Kershaw

reports have been studied by I should be delighted with the Peer

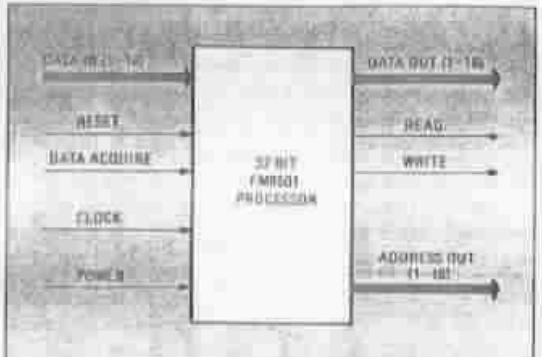
TECHNOLOGY TO WATCH

MATHEMATICAL PROOF VERIFIES ERROR-FREE PROCESSOR DESIGN

ALGORITHM ENSURES THAT 16-BIT CHIP HAS NO HARDWARE BUGS

Proving out a new microprocessor design is as tough a challenge for hardware designers as coding an error-free program is for software engineers. As a result, hardware designers are starting to pick up new tricks, some borrowed from their software counterparts: they are using mathematical techniques to specify, design, and verify chip designs before fabrication. The goal is to catch problems before a chip hits the market.

Though rarely applied in the past, these techniques are gaining popularity and achieving results. The result of one recent effort is Viper, the 32-bit Verifiable Integrated Processor for Enhanced Reliability developed at the UK's Royal Signal and Radar Establishment [*Electronics*, Jan. 27, 1986, p. 51]. Now researchers on this side of the Atlantic have come up with a chip they describe as a formally specified and mechanically verified microprocessor. Using an automated



Collaboration with the Royal Signal and Radar Establishment (RSRE)

- John Cullyer (RSRE) wanted to collaborate with us.
- A memorandum of understanding was completed and NASA Langley agreed to perform some validation studies on the VIPER.
- I asked John Cullyer who he would like to have work with us---He suggested Warren Hunt of Computational Logic.
- We also contracted with Boeing Commercial who sub-contracted with Karl Levitt now at U.C. Davis

Bishop Brock/ Warren Hunt Report

Report on the Formal Specification and Partial Verification of the VIPER Microprocessor, NASA CR-187540, July 1991

Before we would be satisfied that VIPER was verified in the formal sense, we would expect to see complete formal specifications at every hierarchical level, from the top-level instruction interpreter down to the gate-level design. Accompanying these specifications should be proofs which showed that the gate-level design correctly implements the top-level machine…. We pointed out several of these deficiencies, including the use of the informal simulation language ELLA for the gate-level specification, the lack of rigor in the Intelligent Exhaustion analyses, and the incomplete nature of RSRE's block-level specification. These points, and the fact that the attempt to prove the correspondence between the top-level and block-level machines in HOL is incomplete, **lead us to the conclusion that VIPER has not been formally verified.**

Levels of Rigor

- I appealed to Warren Hunt to soften conclusion based on:

Level 0: Static Code Analysis (No semantic analysis)

Level 1: Specification using mathematical logic (no proofs)

Level 2: Formal Specification + Hand proofs

Level 3: Formal Specification + Mechanical proofs

Level 4: Formally verified prover (a.k.a. rigor mortis)

Water is relatively low but the commercial gain is quite high," says business manager Dave Davies.

EIS is offering a range of software based on the 17 scientific and financial applications, built-up in house over the last four years. It says it will supply packaged software, custom software, planning, training and support. It hopes to attract not only water companies for its services but also large commercial and industrial organisations.

"As we have found to our great benefit IT is as vital to large modern companies as water itself," says Welsh Water chairman John Elfed Jones. "We not only take our own medicine but thrive on it."

EIS, set up in-house in 1988, has a staff of 35 developing applications with ICL, DEC, IBM, Ingres and Oracle products. It says it is committed to open systems and portability is built into each application.

EIS's portfolio covers customer accounts, quality control, energy management, planned maintenance, stock control and work management. Welsh Water claims its energy management system saved the company £1.4m in a single year.

■ Jason Hobby

User threatens court action over MoD chip

The first commercial user of the Viper safety-critical chip developed by the Ministry of Defence is threatening legal action for alleged misrepresentation.

Teknis International Railroad Systems of Adelaide, Australia, is seeking assurances that the Viper technology can meet the claims that the MoD and its commercial partners make for it.

Teknis, which is developing a signal and railway crossing control system using Viper for the Australian National Railway Commission, is also threatening action against the MoD's commercial licensee, Charter Technologies.

Worcester-based Charter was licensed in January 1988 to exploit commercially the fruits of the Viper work carried out at the Royal Signals and Radar Establishment at Malvern.

Ron Davison, Teknis' business development manager, says, "We are looking for every comfort we can get from the development and suppliers of Viper".

Davison says the A$12m Australian railways project "is a world first" in the safety-critical market, marking the first time that Viper has found a user outside the military and defence communities.

Teknis' concern has been inspired by a series of reports in UK and US academic circles about RSRE and Charter's claims that Viper is formally verified for use in safety critical applications where lives may be put at risk if the technology fails.

Davison says he is "surprised at the sudden rash of reports about Viper coming out of the woodwork" 18 months after Teknis began work with the chip.

But the report that is most critical of Viper, written by Avra Cohn of Cambridge University's computer laboratory, is two years old. It was published in May 1988 and delivered to RSRE, but Charter Technologies claims it was not shown Cohn's findings until mid-1989.

RSRE and Charter claim that Viper is formally specified, with a chip design which conforms to this specification. Cohn says in the report that this is misleading.

CULLYER... Backs Nasa.

"Such assertions, taken as assurances of the impossibility of design failure in safety critical applications, could have catastrophic results," Cohn says in the report.

The MoD says, "It is a matter of interpretation of the words used to describe the dependability of Viper. Nothing can be described as absolutely fail safe."

This year a report by US consultants Computational Logic for US space agency Nasa says "Viper has not been formally verified" and lists four deficiencies in RSRE's specification. In a draft copy of the same report dated June 1989, obtained by *Computer Weekly*, the former chief RSRE scientist on the Viper project, John Cullyer, has indicated his agreement with Nasa's conclusions. Cullyer is now Professor of Electronics at Warwick University.

The MoD cannot say whether the Nasa and Cohn reports have been looked at by RSRE staff, but a spokesman says, "Work is continuing to reinforce verification techniques and if a relevant report has been produced then it will be studied by scientists at RSRE."

Marconi Electronic Devices of Lincoln, sub-contracted by the MoD to manufacture Viper hardware circuitry, is reining back on its commitment to the project while it waits for replies from the MoD.

Tony Smith, Marconi Electronic Devices' integrated circuits contract manager, says the company "wanted a discussion with MoD and RSRE about what could be guaranteed for Viper. That meeting was due to take place this year, but the MoD cancelled it. We have still not had that meeting".

Marconi has asked the MoD to respond to the Cohn and Nasa reports, but has not yet received a reply and has not been shown either of the reports, Smith says. The company is making prototype Viper circuits, but has no commercial orders.

The Ministry of Defence would not comment on "confidential or commercial correspondence between it and third parties".

The MoD says, "No Viper chip is known to have failed, but work is continuing to reinforce and improve verification techniques" on Viper, and that "although there are no known faults in the Viper design, an unremitting search for weakness must continue".

■ Simon Hill

The fangs of the VIPER

NATURE VOL 352 8 AUGUST 1991

Donald MacKenzie

VIPER is the first commercially available microprocessor whose design is claimed to have been proven correct. The controversy provoked by the claim reveals fundamental disagreement about the meaning of 'proof'.

COMPUTER systems are increasingly taking on roles where their failure could

had, the court would have been asked to rule on what constitutes mathematical

that gate-level realisations conform to this top-level specification"[4]. But the

VIPER LITIGATION

- In 1991 litigation broke out in Britain over the proof of VIPER
- Charter Technologies Ltd., a firm which had licensed aspects of VIPER technology, took legal action against them:
  - Sales of VIPER had been disappointing.
  - They alleged that VIPER's design had not been proven to be a correct implementation of its specification.
- The MOD contested Charter's allegations
- The case did not come to court because Charter became bankrupt before the High Court could hear it.
- **Court would have had to decide what constitutes a <u>rigorous mathematical proof</u>**
- IRONY: No "bug" had been found in the VIPER and their design had been subjected to an unprecedented amount of testing, simulation, checking and mathematical analysis.

The Outcome

- VIPER project at RSRE was defunct
- The two most famous disasters in formal methods history: SIFT OS and VIPER, and we have been a part of both of them! … agh ...

But we now have research dollars in formal methods and I now had authority to build an in-house team

Competitive Contracts Awarded to Move Formal Methods Into Practice in the Aerospace World.

- In 1990 three contracts were awarded after a competition:
  - Computational Logic
  - Odyssey Research Associates
  - SRI International

GOAL: Apply **existing methods** to real aerospace applications

Finally, Some Recognized Successes:

- Rockwell Collins/SRI Verification of AAMP5/AAMP-FV μPs (Srivas, Miller)

- Proved microcode of one instruction in each instruction class of their new high-performance AAMP5
- Significant errors found:

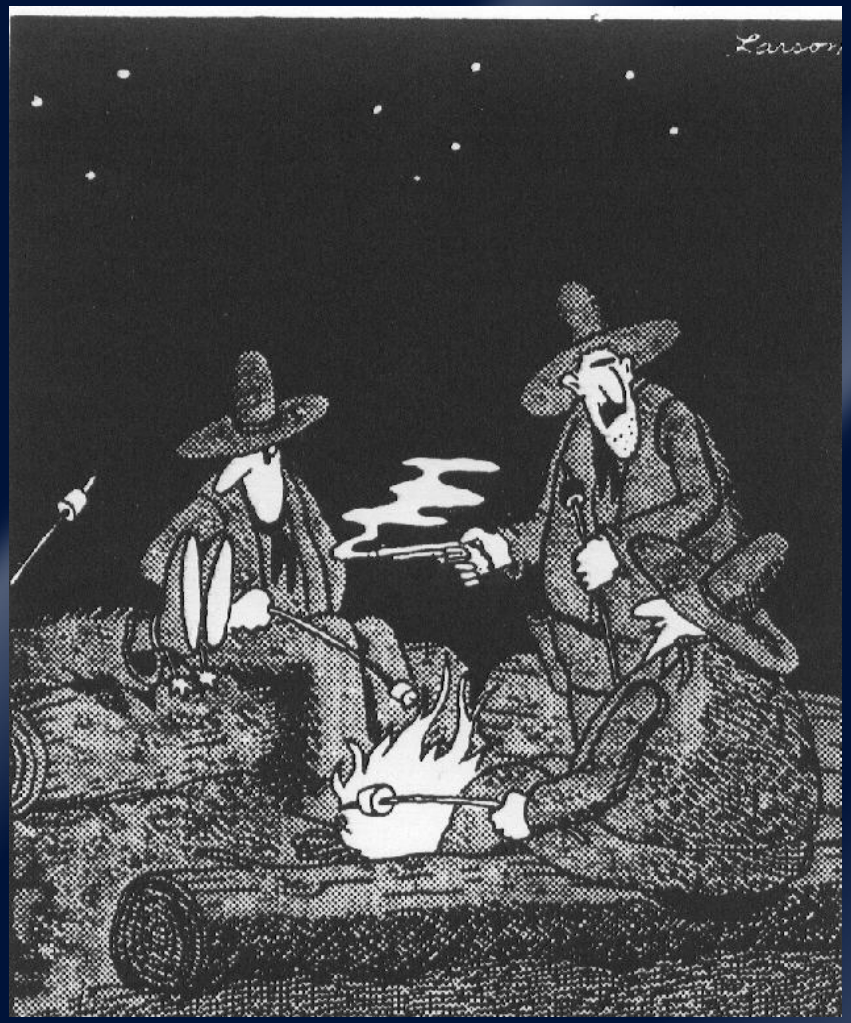
A letter from Charlie Kress, Manager, Processor & Software Technology, Rockwell Collins (Jan 18, 1995) stated:

> While this was an exploratory project, it actually uncovered two errors in the AAMP5 microcode. Moreover, errors that we seeded in the microcode were systematically uncovered by SRI, clearly demonstrating the potential of this approach. *As a direct result of this success, we have committed to work with SRI and NASA next year to formally verify the design of a smaller microprocessor specifically designed for ultra-crucial applications.*

At end of this project:

- There were four engineers at Collins that were skilled in formal methods.
- In fall 1996 Rockwell Collins hired a formal methods expert whose full-time job is to integrate the use of formal methods into their product lines.

We were now committed to theorem prover technology



You guys are both my witnesses…He insinuated that intuitionistic logic is superior to classical logic!

Finally, Some Recognized Successes (cont.)

- Formal Analysis of Fault-tolerance protocols under Allied-Signal's hybrid fault models (Lincoln, Rushby)

  Good
  Benign faulty
  Symmetric faulty
  Asymmetric faulty

- Formal Analysis of Shuttle Software Upgrades (GPS, 3EO) using PVS and model checking and NASA FM guidebook (Di Vito, Crow, Kelly)

A Partial List Of Langley-Funded Projects: 1991 - 2000

- Boeing PIU Project (1991)
- Charles Stark Draper FTPP Scoreboard Project (1991)
- Allied Signal Hybrid Fault Models (1992)
- Space Shuttle Jet-Select Project (1993)
- DDD - Digital Design Derivation (1993)
- Rockwell Collins AAMP5 (1994)
- Union Switch and Signal (1994)
- Honeywell Air Transport Systems (Tablewise) (1995)
- Rockwell Collins AAMP-FV (1995)
- Space Shuttle GPS and 3EO upgrades (1995)
- Integrated Modular Avionics and RTCA SC-182 (1997)
- Collins Mode Confusion Project (1998)
- ORA/Aonix Ravenscar Project (1998)
- Formal Analysis of UML Models (1999)
- Aircraft Information for Lateral Spacing (AILS) (1999)
- PVS: formal semantics, batch, execution engine (2000)

http://atb-www.larc.nasa.gov/rwb-fm-dating-service.html

Formal Methods Dating Service

Are you tired of ambiguous specifications?
Do you long for formally defined semantics?
Do visions of putative theorems dance in your head?
Are you tired of phoney theorem provers lying about your axioms?

Then this site is for you!

We're dedicated to matching desperate engineers with eager formal methodists!

There are two ways to join Rick's service:

- Click **here** to initiate your personal quest for the perfect match for all of your longings
- If you're too desperate to trust the whims of electronic submission, call 1-888-CALL-RWB.

Curator: ***Ricky W. Butler (R.W.Butler@LaRC.NASA.GOV)***

last modified: 11 September 1997

Lfm 97
Michael
Holloway

Some Details about 4 Projects (2000 – 2004)

- DEOS Project -- Under ITSR program
- SPIDER Project -- Under AVST program
- Honeywell Engines and Systems (AvSP)
  TTA-based FADEC (with TTTech and SRI International)
- Rockwell Collins (AvSP)

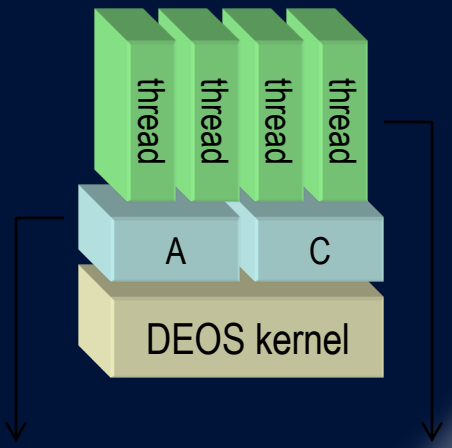
Honeywell Technology Center
with SRI International

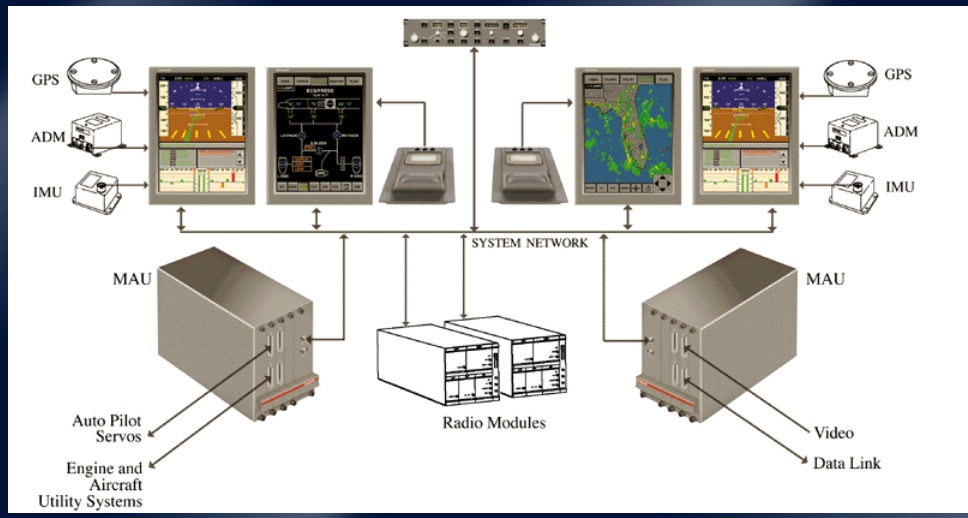
GOAL: Develop and implement verification techniques for demonstrating safety of IMA software using the DEOS operating system as the test subject.

Primus Epic

- DEOS is a partitioned real-time operating system used in Honeywell's Primus Epic developed for DO-178B Level A software
- Preemptive rate-monotonic scheduler
- Mixed criticality tasks

DEOS Project (cont.)





- Formal modeling and verification of time and space partitioning
- Semi-formal techniques for implementation-level correctness

Successful Application of Model Checking to Timing Analysis on early versions of DEOS

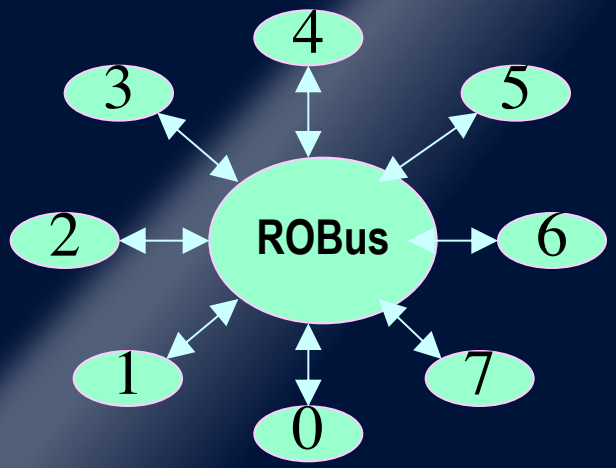
CHALLENGE:

slack-time reclamation

SPIDER

2000 - 2005

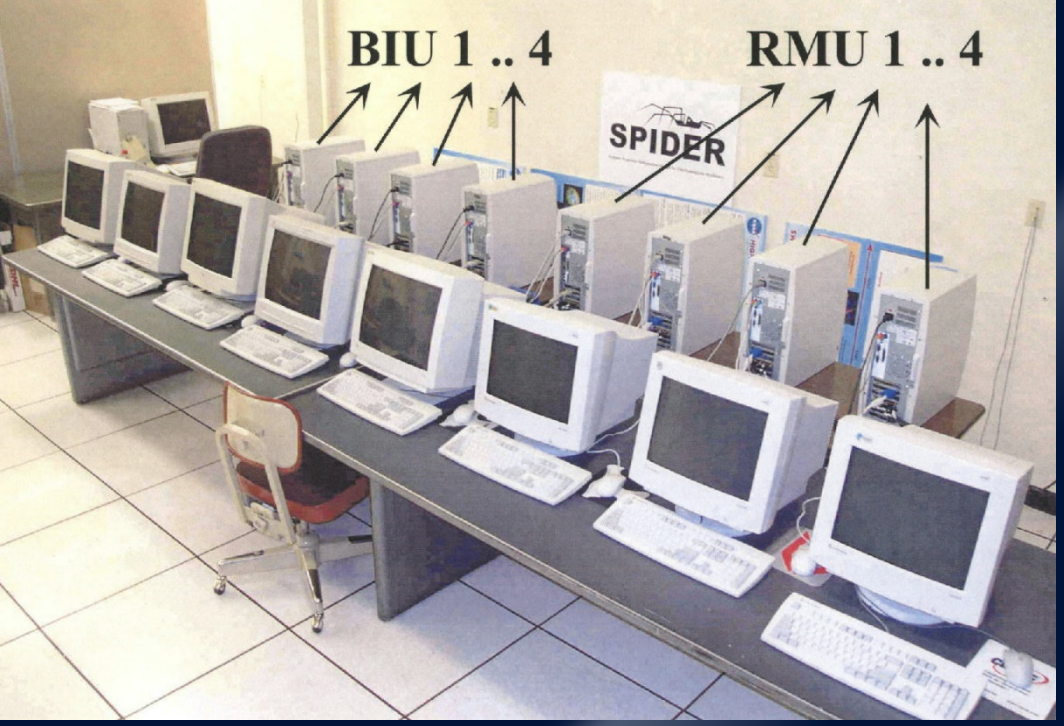
- **SPIDER:** Scalable Processor-Independent Design for Electromagnetic Resilience
- Builds upon 20 years of fault tolerance research at LaRC
- Co-funded by FAA and NASA Langley
- GOALS: Develop fault-tolerant computer architecture in accordance with RTCA DO-254 guidelines:



– demonstrate feasibility of formal methods as means of certification

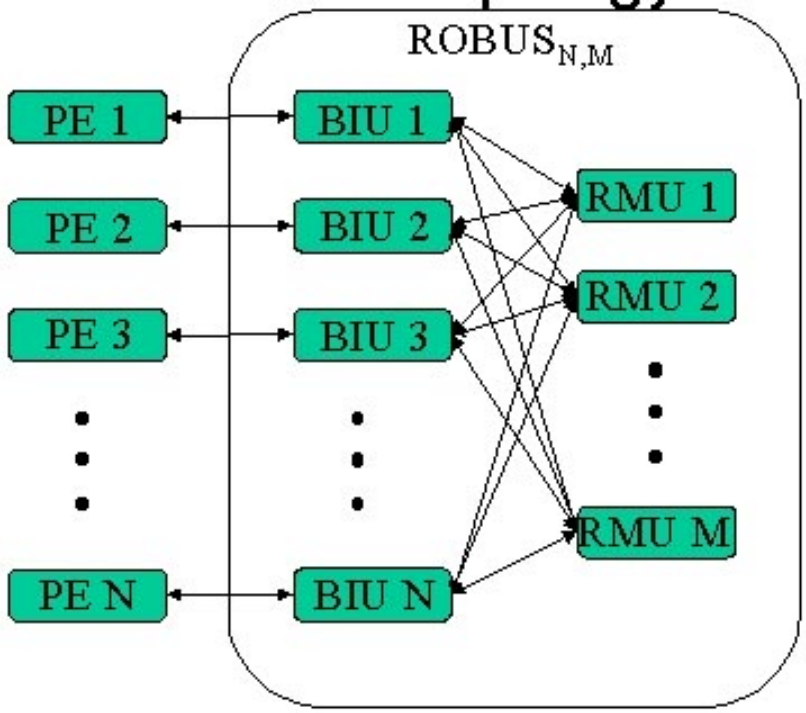
– develop training materials for FAA

– develop advanced fault-tolerant computer architecture platform for inhouse analysis and experimentation



- Bus statically scheduled as in TTA
- Processing elements need not be the same
- **Formally verified group membership**
- **Formally verified clock synchronization**
- **Formally verified interactive consistency** (Byzantine resilience)

- **Reliability analysis** using SURE
  - Calculates P(enough good hardware)
- Formal proof of fault-tolerance protocols using PVS
  - enough good hardware => correct operation



Honeywell Engines and Systems with TTTech and SRI International

GOAL: Develop Fault Tolerant Integrated Modular Architecture design, validation, and implementation technologies for deployment in next-generation engine controls for commercial aircraft.

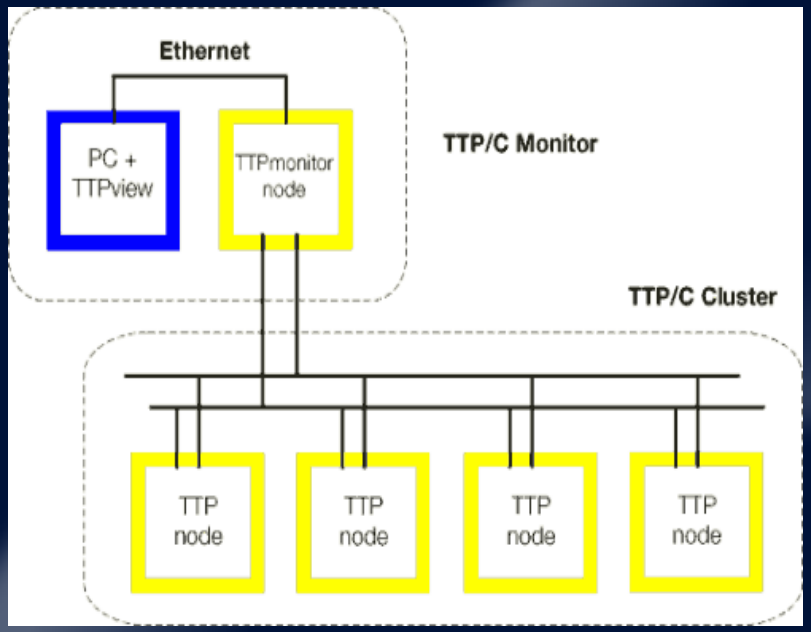
APPROACH: Use TTTech's Time Triggered Architecture (TTA) developed in Europe for the automotive industry and formal verification methods (SRI) to develop a FTIMA architecture. Targeted application is Full Authority Digital Engine Control (FADEC).

Composability
Predictable temporal behavior
Diagnosability and Testing
Reusability of Components
Fault-tolerance

Honeywell Engines and Systems
with TTTech and SRI International

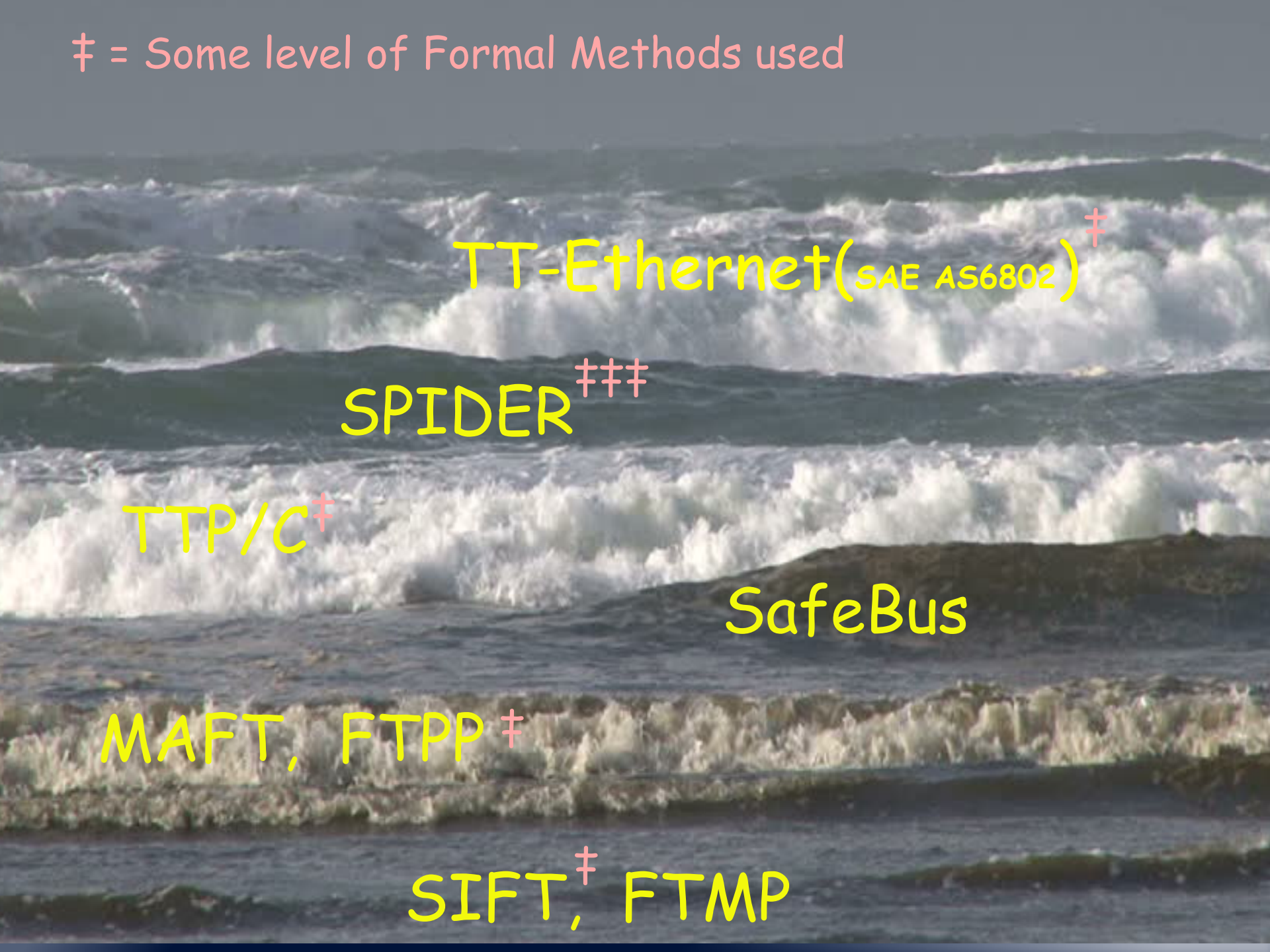
Formal verification of partitioning and fault tolerance and FAA pre-qualification.

"TTTech's cost efficient and highly researched Time Triggered Architecture has been identified as the winning solution. The modular and scaleable TTP-based MAC platform is targeted for many safety critical applications. TTP provides a major competitive advantage for Honeywell in the aerospace industry." [Jim Zerban, Product Line Manager of Electronic Controls and Sensors at Honeywell Tucson]



"TTTech is very excited about the partnership with the originator of the Safebus in the Boeing 777 and the world's leading supplier of flight critical systems. This partnership and joint FAA certification of TTTech's products will ensure that TTP systems comply with the highest safety standards." [Stefan Poledna, CEO of TTTech.]

‡ = Some level of Formal Methods used



The Research "Wave"

As you know, TTEthernet took great benefit of the fundamental concepts that you developed with the NASA SPIDER ROBUS protocol…

… we have been using the formal methods tools from SRI International (pretty much all of them) in the development, analysis, and even configuration of TTEthernet

Dr. Wilfried Steiner - Corporate Scientist
at TTTech Computertechnik AG

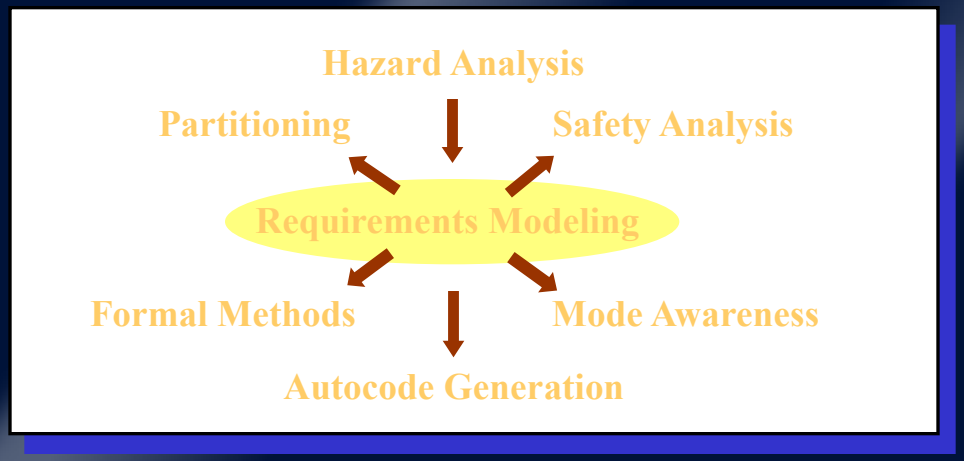
Steve Miller

Advanced Technology
Center of Rockwell Collins

Rockwell Collins Advanced Technology Center

GOAL: Develop formal methods and tools for designing and verifying flight software that can improve safety in the civil fleet.

- Flight Guidance System Modeling and Analysis
- Flight Management System Modeling and Analysis
- Detection of Mode Confusion Potential
- Hazard/Fault Analysis
- Auto-generation of code.



Collins Slide: Then (1999) and Now (2004)

ADVANCED COMPUTING SYSTEMS

Five Years Ago

- Model-Based Development Routinely Dismissed
- Formal Methods Viewed as Impractical & Too Expensive
- Created Models by Hand Using Research Notations
- Verifying Representative Examples - in Weeks
- Tools were Research Prototypes

Today

- Widespread Acceptance
  - *787, FCS 5000, ARJ, MUE, FMS …*
- "This is Buck Rogers!"
  - *actual customer quote*
- Automatically Translate Models from Leading Commercial Tools
- Finding Real Errors in Real Systems - in Seconds
- Tools being Matured for Enterprise Use and Support

Rockwell Collins May 11, 2004 Letter

Raj K. Aggarwal
Vice President, Advanced Tech Center
Advanced Technology Center

Engineering and Technology
400 Collins Road NE
Cedar Rapids, IA 52498-0001 USA
319.295.0053 Fax: 319.295.5085
Cell: 319.573.6643
raggarwal@rockwellcollins.com

June 13, 2003

Rockwell Collins

Dear Rick,

I am writing to express our appreciation for your support of the Methods and Tools for Flight Critical Systems Project. This project is playing an essential role in developing the techniques and tools needed to build the avionics systems of the future. During the past two years, we have demonstrated an impressive array of capabilities, ranging from formally modeling significant portions of a Flight Guidance and Flight Management System, to formally verifying the correctness of the Flight Guidance System mode logic, to using model checkers and theorem provers to identify potential sources of mode confusion. Other researchers have demonstrated these capabilities on toy examples. Through your program we have demonstrated them on examples representative of the systems actually built in industry.

During the past year, the two major airplane companies made model-based development a requirement for winning contracts with them. Because of the groundwork established on this project, subcontractors are able to respond quickly to their requests and demonstrate how they can help them to improve safety while reducing costs and cycle time. By selecting the right methods and tools, building a costly legacy of software based on tools that will be obsolete in a few years can be avoided.

There is a great deal of work that remains to be done. While much of this is cultural change in our industry, research in this field will be needed for many years. Over the next decade, we expect to see formal analysis become as critical to the design of safety-critical systems as testing is today, with manufacturers using their latest methods and tools a key competitive advantage. We also hope to transform system safety analysis from the art it is today into a true engineering discipline.

Rockwell Collins is working to identify and reduce one of the most common causes of civil aircraft accidents, mode confusion. Our Business and Regional Systems division has agreed to team with us in a redesign of the Vertical Navigation function, one of the most confusing avionics functions. They are also working with us to prototype a major change in flight deck design that could significantly reduce the potential for mode confusion.

There is sufficient interest and demand in all these areas that over the last year Rockwell Collins has initiated a large internal IR&D project on model-based development. We are also seeking to leverage NASA's investment by seeking additional external funding to accelerate the development and transfer of these technologies.

We look forward to working with NASA Langley Research Center over the next three years as we complete this project, and to participating in the definition of the next phase of the Aviation Safety Program.

Sincerely,

Raj Aggarwal

Raj Aggarwal, Vice President, Advanced Technology Center

As I'm sure you know, the Methods and Tools for Flight Critical Systems project provided us an opportunity to investigate the use of Model-Based Development in designing software for complex avionics systems. Rockwell Collins has taken the ideas explored in this project and expanded them into a sophisticated suite of tools, methods, and processes that will improve the safety of our systems while also lowering the cost of their development. Our expertise in Model-Based Development played a key role in our recent win of the Displays and Crew Alerting (DCA) system on the Boeing 7E7 Dreamliner."

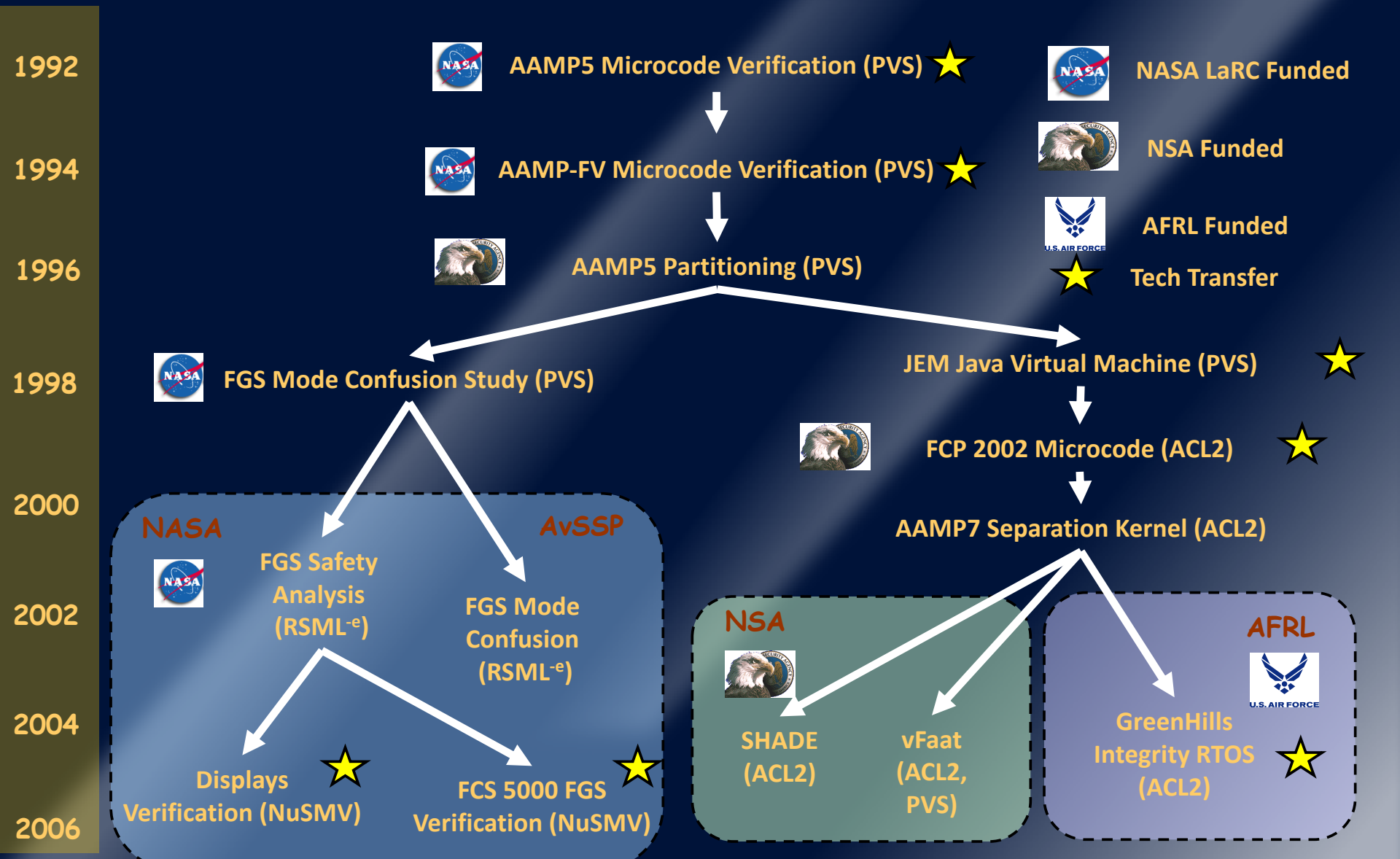
Raj Aggarwal, Vice President, Advanced Technology Center

May 11, 2004 Letter (cont)

"On the 7E7 program, we are investigating the use of model-checking to prove the correctness of the reversion logic in the DCA System Windows Manager…. The reversion logic is a critical system function that poses unique design challenges, and it appears that model-checking will allow us to verify its design to the highest possible level of assurance.

On another project, we are using the formal modeling and analysis capabilities developed under the Methods and Tools project to verify the design of the mode logic for the Flight Guidance System and Autothrottle of our new FCS 5000 product line

Historical Perspective (Collins)



Index of FM Projects

Hardware

Boeing PIU Project (1991) †
Draper FTPP Scoreboard (1991) †
Rockwell Collins AAMP5 (1994) †
Rockwell Collins AAMP-FV (1995) †
Derivation Reasoning System (1998)
Honeywell/TTTech FADEC (2000+)

Software

Space Shuttle Jet-Select Project (1993) †
Honeywell Air Transport (Tablewise) (1995) †
Space Shuttle GPS & 3EO upgrades (1995) †
Collins Mode Confusion Project (1998)
ORA/Aonix Ravenscar Project (1998)
Formal Analysis of UML Models (1999)
Collins AvSP Project (2000+)

Fault-tolerance

Reliable Computing Platform (1990)
Allied Signal Hybrid Fault Models (1992)
Union Switch and Signal (1994) †
SPIDER (2000+)

Systems / Algorithms for ATM

Aircraft Info for Lateral Spacing (AILS) (1999)
Barron--Analysis of Neural Nets (2002)
Conflict Detection and Resolution (2002)
Small Aircraft Trans System (SATS) (2003)
Self-Spacing Terminal Area (2003)

Software Certification

Streamlining SW Aspects Cert (2001)
MC/DC studies and tutorial (2002)
Object Oriented Software and Cert (2003)

Tools and Techniques

PVS Theorem Prover Development
Accident Report Analysis
Zeus: Natural Lang + FM

51

Air Traffic Management

2001 -> Present

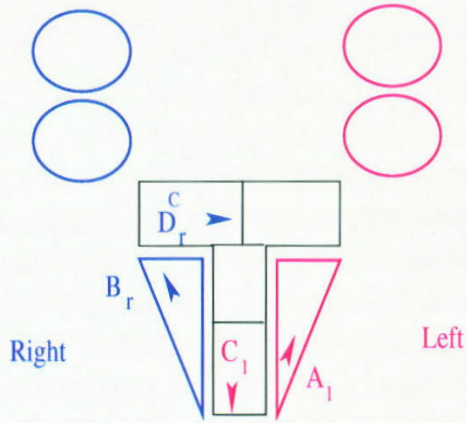
César A. Muñoz

-Arrived 1999 joined the Formal Methods group via ICASE*
- 2003 to 2008, worked for us via the NIA
- 2009 to present: NASA Langley civil servant

Formal Analysis of Small Aircraft Transportation System (SATS)

- SATS goal: significantly increase the capacity of regional airports.
- Use of a **software system** to sequence aircraft into the SATS airspace with no air traffic controller present.
- Formal Methods applied to a concept of operation.

- A formal **finite-state machine model** of the SATS operational procedures (24 transition rules)
- **Exhaustive analysis** of entire state space
- **Six Safety Properties** verified including
  - At most one aircraft cleared at a given fix
  - There is always a MAHF for every aircraft
  - No more than 2 aircraft on missed approach for a given fix.
  - Runway incursions do not occur
- Liveness properties verified, (e.g. no deadlocks)



(b) Aircraft B is faster than A

Formal Analysis Of SATS Concept of Operations

Operational procedures captured in 24 formal transition rules.

- Nine issues identified via analysis
  two required changes to the rules of the ConOps, five where implicit or explicit omissions, and two were clarifications.
- All recommendations from FM team adopted by SATS conops team

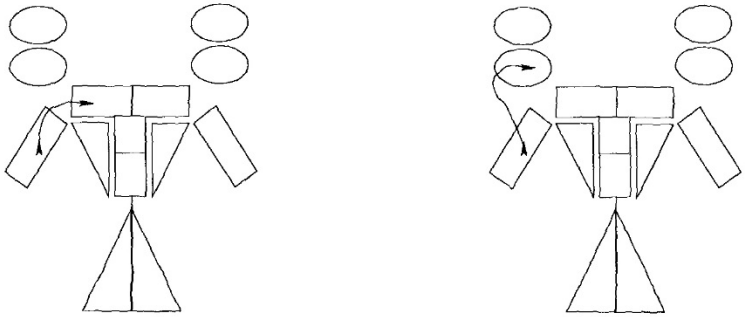
Example:

3.2.5 Approach Initiation for Lateral Entry (right, left)

The Approach Initiation for Lateral Entry (right) procedure is illustrated in Figure 10. An aircraft in lateral entry is allowed to initiate the approach only if the following conditions hold:

- It is the first aircraft in the sequence or its leader is already on the approach.
- There is at most one aircraft on base at the opposite side.

If one of these conditions is not met, the aircraft must hold at 2000 feet.
This procedure is encoded by the PVS function `LateralApproachInitiation`.



Leader on the approach and enough spacing Otherwise

Figure 10: *Approach Initiation for Lateral Entry (right)*

```
VerticalEntry(side)(this):list[SCA] =
  IF virtual(this,side) < 2 &
    NOT on_approach?(this,side) &
    length(this`maz(side)) = 0 &
    length(this`lez(side))  = 0 &
    length(this`holding3(side))  = 0 THEN
   LET a = aircraft(this,side) IN
   LET next = this WITH [
     `holding3(side):= add(this`holding3(side),a),
     `nextseq      := next(a),
     `nextmahf     := opposite(a`mahf),
     `nextid       := this`nextid+1,
     `rule         := 1*sign(side)
   ] IN
   (: next :)
  ELSE
   null
  ENDIF
```

State-based Conflict Detection and Resolution Algorithms

Formally Verified Algorithms

KB3D -- pairwise (1991) [Dowek, Geser, Munoz]

ACCORD – formally proved implicit coordination (2005)

-- formally verified algorithms that recover from loss of separation (2008)

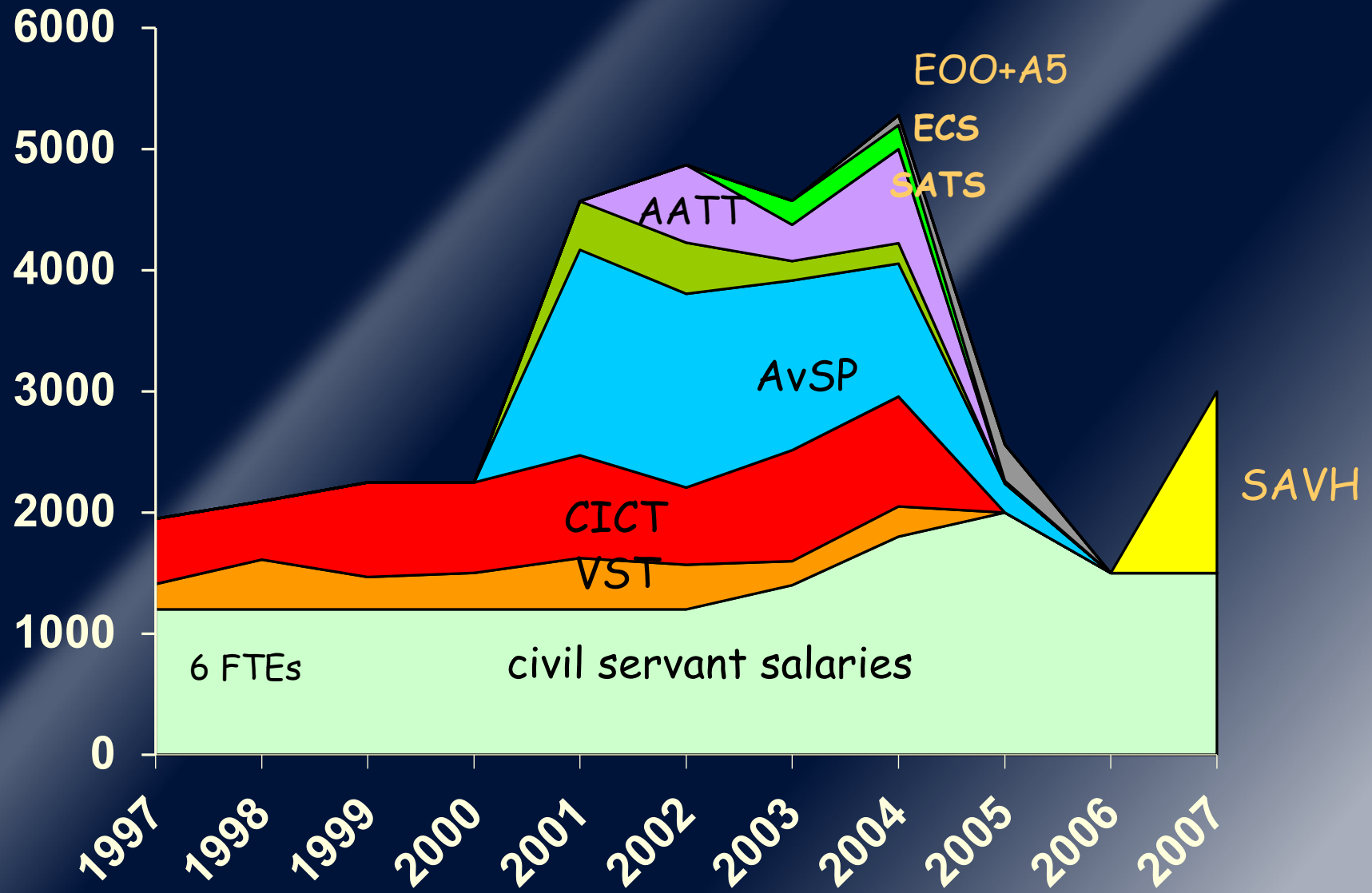
Bands – formal verification of Prevention Bands (2010)

Chorus – based on formal criteria for implicit coordination (1 to N algorithm) (2013)

Just two more lemmas and the proof of the KB3D CD&R algorithm will be complete!!!!

Lisa J. Porter was the NASA Associate Administrator for the Aeronautics Research Mission Directorate in 2005 -- 2008

Formal Methods Budget (1997-2007)



New Porter era – radical change to our procurement:

- NRAs controlled at NASA Headquarters level

- Aeronautics budget cut in half to support Project Constellation

Formalization of “Well clear”

Motivation for A Formal Model of "Well Clear" for Unmanned Aircraft

On-board pilots have responsibility for "not operating an aircraft so close to another aircraft as to create a collision hazard" [International Civil Aviation Organization (ICAO) 2005a; US Code of Federal Regulations 1967a]

"to see and avoid other aircraft" [International Civil Aviation Organization (ICAO) 2005b; US Code of Federal Regulations 1967b],

and when complying with the particular rules addressing right-of-way, on-board pilots "may not pass over, under, or ahead [of the right-of-way aircraft] unless well clear" [International Civil Aviation Organization (ICAO) 2005b; US Code of Federal Regulations 1967b].

Motivation for A Formal Model of "Well Clear"

- In 2012 there was a very informal notion of "well clear" that was in the regulations
- There was no mathematical model.
- It is probably a good idea not to set off a TCAS alert, so "well clear" should contain the TCAS volume.
- But there was no mathematical model of TCAS!
- So Cesar Munoz and Anthony Narkawicz created one:
- César Muñoz, Anthony Narkawicz, and James Chamberlain, *A TCAS-II Resolution Advisory Algorithm*, Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC), AIAA-2013-4622, Boston, Massachusetts, August 2013.

So the Langley team created a mathematical definition of “well-clear”

Anthony Narkawicz, César Muñoz, Jason Upchurch, James Chamberlain, and María Consiglio, A Well-Clear Volume Based on Time to Entry Point, Technical Memorandum, NASA/TM-2014-218155, January 2014.

Jason Upchurch, César Muñoz, Anthony Narkawicz, James Chamberlain, and María Consiglio, Analysis of Well-Clear Boundary Models for the Integration of UAS in the NAS, Technical Memorandum, NASA/TM-2014-218280, June 2014. BibTeX Reference.

César Muñoz, Anthony Narkawicz, James Chamberlain, María Consiglio, and Jason Upchurch, A Family of Well-Clear Boundary Models for the Integration of UAS in the NAS, * Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA-2014-2412, Atlanta, Georgia, 2014

that contains the TCAS region

And nice, verified mathematical properties:

Symmetry: in a pairwise scenario both the ownship and intruder aircraft agree on well-clear status

Inclusion: the well-clear model based on time to entry point is more conservative than tau, time to closest point of approach, and modified tau for any scenario and any common choice of threshold values

Local Convexity: from the point of view of the ownship, any ray that points towards the violation area has only one intersecting segment.

The mathematical formulas and theorems for "Well Clear" were formally specified and verified in the Prototype Verification System (PVS)

against a time threshold. In this paper, the time threshold is called TTHR. An example of a time variable that is used in conflict detection logics is t_{cpa} [4].

The time variable used in earlier versions of the TCAS detection logic is called *tau*, denoted τ [8]. Tau estimates t_{cpa}, but is less demanding on sensor and surveillance technology than t_{cpa}. Indeed, τ is simply defined as range over closure rate, where closure rate is the negative of the range rate, i.e., $\tau = -\frac{r(0)}{\dot{r}(0)} = -\frac{\|\mathbf{s}\|}{\frac{\mathbf{s}\cdot\mathbf{v}}{\|\mathbf{s}\|}} = -\frac{\mathbf{s}^2}{\mathbf{s}\cdot\mathbf{v}}$. This paper defines τ as -1 when the aircraft are not horizontally converging. Formally,

$$\tau(\mathbf{s},\mathbf{v}) \equiv \begin{cases} -\frac{\mathbf{s}^2}{\mathbf{s}\cdot\mathbf{v}} & \text{if } \mathbf{s}\cdot\mathbf{v} < 0, \\ -1 & \text{otherwise.} \end{cases} \tag{6}$$

For a limited number of scenarios, the values of τ and t_{cpa} coincide. However, in most scenarios, the value of τ tends toward infinity as the aircraft approach the closest point of approach. In general, τ is a good approximation of t_{cpa}, but only for large values. For that reason, TCAS II uses a modified variant of τ called *modified tau*, denoted τ_{mod} [8]. Modified tau provides a better estimation of t_{cpa} and has a more desirable behavior than τ in the proximity of the closest point of approach. In [3], modified tau is defined such that $\tau_{mod} = -\frac{r(0)^2 - \mathtt{DTHR}^2}{\dot{r}(0)} = \frac{\mathtt{DTHR}^2 - \mathbf{s}^2}{\mathbf{s}\cdot\mathbf{v}}$. Similar to τ, τ_{mod} is defined as -1 when the aircraft are not horizontally converging, i.e.,

$$\tau_{mod}(\mathbf{s},\mathbf{v}) \equiv \begin{cases} \frac{\mathtt{DTHR}^2 - \mathbf{s}^2}{\mathbf{s}\cdot\mathbf{v}} & \text{if } \mathbf{s}\cdot\mathbf{v} < 0, \\ -1 & \text{otherwise.} \end{cases} \tag{7}$$

The definition of τ_{mod} in Formula (7) depends on DTHR, which is a horizontal distance threshold. This threshold is called *DMOD* in the TCAS II RA logic, and its actual value depends on a sensitivity level based on the ownship's altitude [8].

In [6], a time variable called *time to entry point*, denoted t_{ep}, is proposed. Time to entry point is defined as the time to loss of horizontal separation with respect to DTHR assuming straight-line aircraft trajectories. Similar to t_{cpa}, t_{ep} decreases linearly over time. Time to entry point is formally defined as

$$t_{ep}(\mathbf{s},\mathbf{v}) \equiv \begin{cases} \Theta(\mathbf{s},\mathbf{v},\mathtt{DTHR},-1) & \text{if } \mathbf{s}\cdot\mathbf{v} < 0 \text{ and } \Delta(\mathbf{s},\mathbf{v},\mathtt{DTHR}) \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{8}$$

where

$$\Theta(\mathbf{s},\mathbf{v},D,\epsilon) \equiv \frac{-\mathbf{s}\cdot\mathbf{v} + \epsilon\sqrt{\Delta(\mathbf{s},\mathbf{v},D)}}{\mathbf{v}^2}, \tag{9}$$

$$\Delta(\mathbf{s},\mathbf{v},D) \equiv D^2\mathbf{v}^2 - (\mathbf{s}\cdot\mathbf{v}^{\perp})^2. \tag{10}$$

The function Θ is only defined when $\mathbf{v} \neq \mathbf{0}$ and $\Delta(\mathbf{s},\mathbf{v},D) \geq 0$. In this case, it computes the times when the aircraft will lose separation, if $\epsilon = -1$, or regain separation, if $\epsilon = 1$, with respect to D. When the aircraft are not horizontally converging or $\Delta(\mathbf{s},\mathbf{v},D) < 0$, time to entry point is defined as -1. Formula (8) is well defined since the condition $\mathbf{s}\cdot\mathbf{v} < 0$ guarantees that $\mathbf{v} \neq \mathbf{0}$.

In 2013, the RTCA organization established Special Committee 228 (SC-228) to define the minimum operational performance standards for a UAS sense and avoid concept, based on “well clear” .

The NASA Langley formal model was chosen and the Langley Formal Methods group is currently participating in the RTCA SC-228. The team was given the responsibility for the specification, development, and verification of a reference implementation of the algorithms that support the overall UAS DAA concept.

Note: Details are Important

Some teams have implemented their own version of “well-clear” but using different parameters.

But, the local convexity property only holds for certain combinations (delineated in the formal proofs)

Many of these other systems have not behaved properly – they should have used formal methods ☺

Cesar and FM team can tell you many stories of how they have had to provide counter-examples and graphics to show why the other variants were not wise choices.

For example, the Well Clear algorithms developed by the DoD were shown to have subtle, but serious flaws after years of development.

The DAIDALUS effort has created an alternative version which has been mathematically proven to satisfy the Well Clear requirement.

DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems

DAIDALUS

Chosen by SC-228 as its official reference implementation of **detect and avoid** for the integration of **UAS** into civil airspace.

DAIDALUS is included in the SC-228 MOPS Document DO-365

Formally verified core algorithms that:

- Determine the current pairwise well-clear status (Detection Logic).
- Compute maneuver guidance to maintain or regain well-clear status (Determine Processing Logic).
- Determine alert type (Alerting Logic).

DAIDALUS core algorithms have been implemented in Java and C++ (≈ 44k lines of code).

Highly configurable interface:

- Aircraft performance limits (acceleration, turn rate, etc.)
- Wind information (simple wind-field model)
- Alerting and guidance thresholds

Code is released under NASA Open Source Agreement (Invention Disclosures LAR-17785-1, LAR-17878-1, LAR-18464-1):

http://github.com/nasa/wellclear.

SC-186 Compact Position Reporting

The Compact Position Reporting (CPR) algorithm is a safety-critical element of the ADS-B protocol: encodes/decodes aircraft position data .

Reports from pilots and manufacturers indicated that some implementations are inaccurate → An FAA Official (Don Walker) suggested that our FM group investigate. (Nov. 2015).

Formal analysis of the CPR algorithm at NASA Langley.

- A formal proof was developed that showed that the published requirements for decoding are insufficient, even if performed using exact real arithmetic.
- A set of new tightened requirements were developed and proved to be correct under exact real arithmetic.
- Mathematically equivalent, but computationally simpler equations were discovered.

Aaron Dutle, Mariano Moscato, Laura Titolo, Aaron Dutle, Cesar Munoz

All suggested recommendations are being considered for the next revision of DO-260.

Very few tools available to do formal analysis of floating point programs.

Led to collaboration with Frama-C group.

Several papers documenting the formal analysis are in development

Formal Analysis

Discovered and formally proved more numerically stable versions of several expressions used in CPR.

- In computation of transition latitudes:

$$\sqrt{\frac{1-\cos(\pi/30)}{1-\cos(2\pi/NL)}} = \frac{\sin(\pi/60)}{\sin(\pi/NL)}.$$

- In encoding (also applies to longitude):

$$\left\lfloor 2^{nb}\frac{MOD(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right\rfloor = \left\lfloor 2^{nb}\frac{lat}{Dlat_i} + \frac{1}{2}\right\rfloor - 2^{nb}\left\lfloor \frac{lat}{Dlat_i}\right\rfloor.$$

- In local decoding (also applies to longitude):

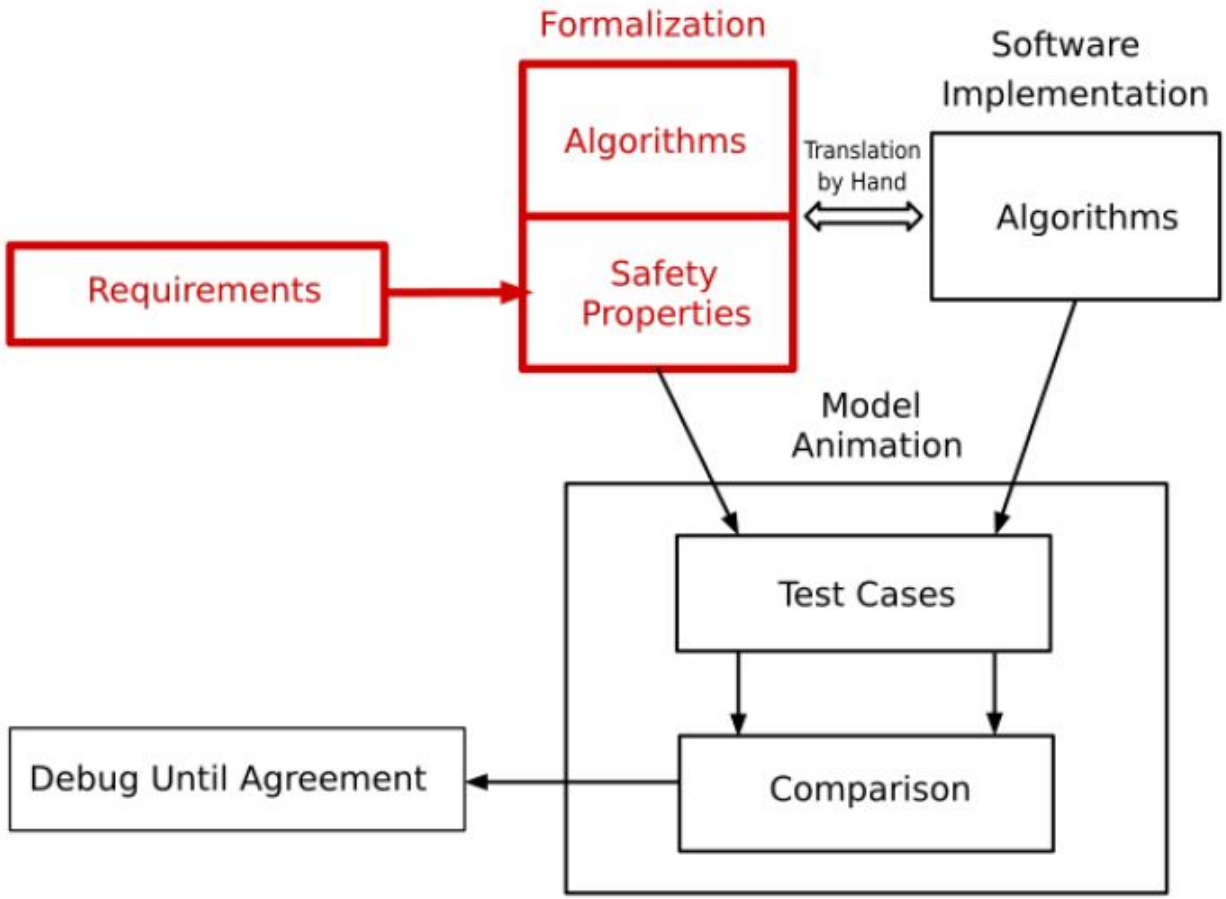
$$\left\lfloor \frac{lat_s}{Dlat_i}\right\rfloor + \left\lfloor \frac{MOD(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{17}} + \frac{1}{2}\right\rfloor$$

$$= \left\lfloor \frac{lat_s}{Dlat_i} - \frac{YZ_i}{2^{17}} + \frac{1}{2}\right\rfloor.$$

MINERVA

Mirrored Implementation Numerically Evaluated against Rigorously Verified Algorithms

MINERVA



Mirrored Implementation Numerically Evaluated against Rigorously Verified Algorithms (MINERVA)

A method for detecting human error in translating from Formal Verified Models to numerical implementation.

- Identify or build a (large) collection of test cases for the software
- Evaluate the test cases in both the implementation and the formal model.
- Compare the results of the two until agreement is reached.

Some Thoughts about Government Sponsored Software Research

- It is absolutely essential that research be performed in close proximity (preferably collaborative) with an actual industrial applications
  - research itself is better:
  - industry challenges often much greater than academic ones
  - research actually has a chance of being used in the real world

Real Applications Push Tool Developers

Our program has driven the development of PVS, one of the most widely used theorem provers in the world

Owre, Rushby, Shankar, Von Henke: **Formal Verification for Fault-Tolerant Architectures Prolegomena to the Design of PVS**, IEEE Transactions on Software Engineering, vol. 21, no. 2, Feb. 1995, pp. 107--125.

- judgements
- new strategies
- abstract data types
- execution engine
- PVS validation
- theory interpretations
- prelude/libraries
- nonlinear algebra
- pragmas

Some Closing Thoughts about Formal Methods Research

Closing Thoughts

Finding great partners is essential!

- Then ask yourself “how can I help them meet their objectives?”

https://insights.rockwellcollins.com/category/formal-methods/

Closing Thoughts

- We have made excellent progress in applying formal methods to many levels of the design process for a large set of domains
- But, there is still a significant gap in connecting our formalisms to the code executing in an embedded system that uses floating point calculations, especially in domains where transcendental functions are used (e.g. Navigation)

Closing Thoughts

Breakthroughs in “Automated Deduction” can have exponential impact on applications

But, program managers are usually not excited about funding foundational work like this

But this can be done within application projects if the COTR is resourceful and committed to it.

Closing Thoughts

It is critical that the funding agencies really understand that simulation and testing cannot establish safety

and develop high-level program plans based on this reality!

Sometimes what looks like a failure can lead to a great success

One of the most successful FM tech transfers

CAUSED by collapse of CLI in 1997

- David Russinoff -> AMD
- Warren Hunt -> IBM
- Matt Kaufmann -> Motorola


Nevertheless,
I would never have dreamed 30 years ago how widely accepted formal methods would become today!
ACL2
SPIN
Coq
HOL
SMT
PVS
NuSMV

THE END

But still the greatest thing about Formal Methods is …

Oh Yeah, my Dad can instantiate higher-order logic formulas with 2000 free variables at the same time!

L–83–6,962


SOFTWARE IMPLEMENTED FAULT-TOLERANCE COMPUTER
ENGINEERING MODEL
INITIAL TASK SCHEDULES
FAULT DETECTION
SOFTWARE VOTE
FAULT
RECONFIGURATION
BY EXECUTIVE S/W
ASSIGNMENT OF
FAULTY UNIT'S TASKS
RECONFIGURED TASK SCHEDULES
SIFT
COMPUTER
SIFT SYSTEM CONTROL PANEL
DATA GENERAL

A Partial List Of FM Projects (2000 – 2004)

- - Streamlining Software Aspects of Certification (2000)
- SPIDER Project (2000 - present)
- New MCDC Testing Method for Software Certification (2001)
- SAL: Model Checking Multi-threaded Java Programs [Stanford Univ] (2001)
- SRI Mathworks Project (2000 - 2002)
- Formal Analysis of Conflict Detection and Resolution Algorithms (2001->)
- Open PVS / ICS : SRI International + Honeywell (2000 -- now)
- Timing Analysis by Model Checking: ORA (2000 - 2002)
- Enhanced SpecTRM-RL tools (2000 ->)
- Rockwell Collins Requirements Analysis/Mode Confusion (2000 ->)
- Honeywell Engines and Systems TTA-based FADEC (2000 ->)
- Barron/Goodrich: certification of non-adaptive neural nets (2000 ->)
- Formal Verification of Self-Spacing and Merging Algorithms (2002 ->)
- University of Virginia/Litton: natural language + FM (2000 - 2002)
- Theory Interpretations in PVS (2002)
- Formal Analysis of Accident Reports (2002->)

Early Formal Methods Research

"John, I just received the final report on the formal proof of the Quicksort function!"

SAFEGUARD

"Think of it like an invisible dog fence, except for drones. Safeguard makes sure that you don't fly into a region, or even a building, that you're not supposed to" [Kelly Hayhurst]

Safeguard is based on formally verified polygon algorithms that predict impending boundary violations via trajectory estimation, and a system architecture that enables certification.